

Lab 06 Template – Ethan Roepke

Part 01

1. Upload python code for babyFeistel cipher (10 points)
 - a. uploaded
2. Final 4-bit ciphertext (5 points)
 - a. 1011
3. Screenshot of encrypting (5 points)

```
cpre3310@cpre3310:~/homework/lab06/part01$ python3 part01_skel.py
4-bit input to encode: 1010
4-bit key: 0111
Round: 0      1110
Round: 1      1101
Round: 2      1011
cpre3310@cpre3310:~/homework/lab06/part01$
```

Part 02

4. Include a screenshot of the two keys in your lab report. (10 points total)

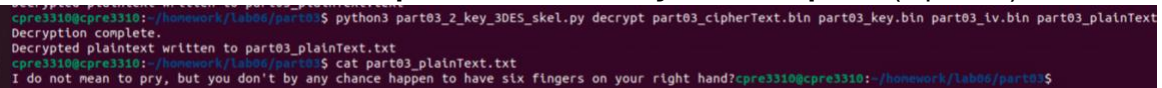
```
cpre3310@cpre3310:~/homework/lab06/part02$ python3 part02_mitm_2DE
Starting MITM attack...
Matching intermediate value found!
Found Key 1 (stripped parity): 0000000000000202
Found Key 2 (stripped parity): 0000000000000200
Keys written to recoveredKeys.txt
cpre3310@cpre3310:~/homework/lab06/part02$ cat recoveredKeys.txt
Key 1: 0000000000000202
Key 2: 0000000000000200
cpre3310@cpre3310:~/homework/lab06/part02$
```

5. What block cipher mode is being used? What is a known flaw with this mode? (10 points total, 5 points each question)
 - a. We are using an Electronic Codebook mode since each block of plaintext is independently encrypted with same key. A flaw with the ECB mode is that it does not include any randomness. Having multiple identical plaintexts that are encrypted will produce identical ciphertexts which will allow attackers reveal patterns.
6. Why is known plaintext important in a meet-in-the-middle attack? What role does it play in recovering the encryption keys? (10 points total)
 - a. Known plaintext is important in a meet-in-the-middle attack to help attackers recover encryption keys when dealing with ciphers that use multiple layers and keys for encryption. The attacker would first encrypt the plaintext with all the key options and store the halfway point in the encryption. The attacker would next take the encrypted ciphertext and decrypt with all the key options. Matching both of the steps an attacker can match the correct keys used. The role for the plaintext is limiting the size of the key search and match the intermediate values.

7. **Could this attack be extended to 3DES? What differences or additional challenges would arise in attacking 3DES with a similar approach?**(10 points)
- Yes this attack can be extended to 3DES but the key length and key space will significantly grow making it more challenging. The attacker would have to store intermediate results for an extra encryption layer. The differences using this type of attack for 3DES is now you are computing 2 layers of encryption and 1 layer of decryption. This requires more computing and comparing multiple stages of encryption. The process different from 2DES is the attacker will do encrypt, decrypt and lastly encrypt.

Part 03

8. **Include a screenshot of the plaintext result in your lab report.** (5 points)



```
cpre3310@cpre3310:~/homework/lab06/part03$ python3 part03_2_key_3DES_skel.py decrypt part03_cipherText.bin part03_key.bin part03_iv.bin part03_plaintext
Decryption complete.
Decrypted plaintext written to part03_plaintext.txt
cpre3310@cpre3310:~/homework/lab06/part03$ cat part03_plaintext.txt
I do not mean to pry, but you don't by any chance happen to have six fingers on your right hand?cpre3310@cpre3310:~/homework/lab06/part03$
```

9. **What block cipher mode are you using in this implementation of Triple DES?**(5points)
- The cipher mode being used in this implementation of Triple DES is the Cipher Block Chaining
10. **In 2-key 3DES, what is the effective key length for encryption? How does it compare to the effective key length of 3-key 3DES? What are those other bits used for?** (15 points total, 5 points each question)
- Each key length in 56 bits, since we have 2 keys being used in the 3DES, we have an effective key length of 112 bits. Compared to a 3-key 3DES, this has 3 keys and therefore has an effective key length of 168 bits. The other bits used for in the 3 key 3DES is used to add complexity and security from attacks.
11. **Historically, there has been a way to use a single key in Triple DES where you set key1=key2=key3. Why would you want to do that? What is the effective key size when key1=key2=key3?**(15 points total, 5 points each question)
- You would want to use a single key in 3DES because older systems that rely on single DES allows to continue function properly and be able to upgrade to 3DES without having to redesign the encryption process. The effective key size for key1=key2=key3 is only 56 bits. You go from encrypt, decrypt, encrypt which is considered a single DES.